AD-A077 169 AIR FORCE AERO PROPULSION LAB WRIGHT-PATTERSON AFB OH F/G 1/2

A COMPUTER PROGRAM FOR ESTIMATING AIRCRAFT LANDING DISTANCE.(U)

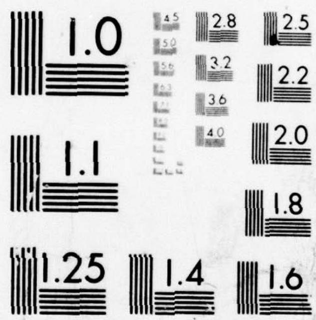
AUG 79 K D MACH

UNCLASSIFIED AFAPL-TR-79-2086 NL

| OF |

AD
A077169

END
DATE
FILMED
12-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

14 AFAPL-TR-79-2086

12

LEVEL II

6 A COMPUTER PROGRAM FOR ESTIMATING AIRCRAFT LANDING DISTANCE

9 Interim rept.
Aug 77 - Jul 79

COMPONENTS BRANCH
TURBINE ENGINE DIVISION

10 Kervyn D. Mach

11 AUGUST 1979

12 49

16 3066

17 06

TECHNICAL REPORT AFAPL-TR-79-2086
Interim Report for Period August 1977 to July 1979

Approved for public release; distribution unlimited.

AIR FORCE AERO PROPULSION LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

012570

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Kervyn D. Mach
KERVYN D. MACH
Aerospace Engineer
Components Branch

Marvin A. Stibich
MARVIN A. STIBICH
Acting Chief
Components Branch

FOR THE COMMANDER

Ernest C. Simpson
ERNEST C. SIMPSON
Director
Turbine Engine Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFAPL/TBC *,W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFAPL-TR-79-2086	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A COMPUTER PROGRAM FOR ESTIMATING AIRCRAFT LANDING DISTANCE		5. TYPE OF REPORT & PERIOD COVERED INTERIM REPORT, AUGUST 1977 - JULY 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Kervyn D. Mach		8. CONTRACT OR GRANT NUMBER(s) In-house
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Aero Propulsion Laboratory (AFAPL/TBC) Air Force Wright Aeronautical Laboratories Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS P.E. 62203F, Proj 3066, Task 06, Work Unit 02
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Aero Propulsion Laboratory (AFAPL/TBC) Wright-Patterson AFB OH 45433		12. REPORT DATE AUGUST 1979
		13. NUMBER OF PAGES 44
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

STOL, Landing Distance, Computer Programs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a computer program which will estimate the runway length required to land a specified airplane. The program accepts inputs describing the aircraft gross weight, velocity, drag, and engine performance and computes the landing distance on specified runway surfaces. Sample results for a time-engine STOL transport are included. The computer program is described along with instructions for its use.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

FOREWORD

This report describes work conducted within the Air Force Aero Propulsion Laboratory, Turbine Engine Division, Components Branch (TBC), Wright-Patterson Air Force Base, Ohio. The work was accomplished under Project 3066, "Gas Turbine Technology," Task 06, "Turbine Technology," Work Unit 02, "Turbine Aeromechanical Analysis," between August 1977 and February 1979. This report was submitted by the author in July 1979.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

I. INTRODUCTION

One aspect of designing an airplane or evaluating the performance of a proposed design is estimating the landing distance it will require. In particular, some airplanes must land on unprepared fields, e.g., wet grass or ice, as well as concrete. The method described in this report was developed to estimate landing distances on these different surfaces. Sample results for a STOL transport are presented and a computer program implementing the method is shown in the appendix.

2. ANALYSIS

2.1 Derivation of Equations

2.1.1 Aircraft Motion

The basic equation is simply

$$F = ma \tag{1}$$

and F is composed of aerodynamic drag, the braking force, and the contribution from the thrust reverser. Thus,

$$F = F_{dr} + F_{br} + F_{rh} \tag{2}$$

where the drag force is a function of velocity and the braking force is the product of the friction coefficient of the tires and the downward force on them. The downward force in turn is composed of the aircraft weight plus any vertical component from the thrust reverser. The reverse thrust is equal to a reverse thrust coefficient times the engine net thrust, and net thrust depends on forward velocity and engine speed. Finally, engine speed varies with time. In equation form

Drag: $F_{dr} = f(V)$ (3)

Braking Force: $F_{br} = C_f * (GRWT + Frv)$ (4)

Horizontal Component of Reverse Thrust:

$Frh = Cv\ rev * Fgross$ (5)

Vertical Component of Reverse Thrust:

$Frv = Frh \tan \theta$ (6)
Where θ is the discharg angle.

Gross Thrust: $Fgross = f(RPM, V)$ (7)

The forces are shown in Figure 1 and the braking coefficients are listed in Table 1. Any moment generated by Frh is ignored.

2.1.2 Engine Dynamics

The engine is assumed to have only two throttle positions - idle and maximum. Its thrust responds to a change in throttle setting according to a first order log equation,

$$F = F_{max} - (F_{max} - F_{idle})\, e^{-t/\tau} \tag{8}$$

where t is the elapsed time in seconds and τ is a characteristic time. At $t/\tau = 5$, the transient is more than 99% complete. Idle thrust is assumed to be zero so that

$$F = F_{max}\,(1 - e^{-t/\tau}) \tag{9}$$

TABLE 1

Braking Coefficients

Surface	C_f
Ice	0.05
Wet Dirt	0.15
Dry Dirt	0.25
Dry Concrete	0.30

2.2 Solution Method

2.2.1 Aircraft Motion

Forward velocity is obtained by numerically integrating equation (1) in a one-term Taylor series. By definition, acceleration $a = \frac{dV}{dt}$, thus

$$\frac{dV}{dt} = \frac{F}{m} \tag{10}$$

$$V_{(t + \Delta t)} = V_t + \frac{F}{m}\,\Delta t \tag{11}$$

The time step, Δt, is taken to be 0.1 seconds.

2.2.2 Assumptions and Boundary Conditions

The aircraft is assumed to touch down with an initial velocity Vo and the engines idling. At time T_1 (typically 1/2 second), the brakes are applied, the thrust reverser is deployed, and the throttle is opened fully. Full reverse thrust is attained after 5 more seconds, at time T_2. Some time later, the airplane's velocity drops below a preset value V_3 (typically 20 ft/sec) and the throttle is closed and the thrust reverser stowed. This time is denoted as T_3. The brakes are applied until the airplane is completely stopped.

The thrust reverser applies both a horizontal and a vertical force to the airplane, as shown in Figure 1. For the particular configuration used as an example here, the resultant reverse thrust vector was inclined at just over 57 degrees, or very nearly one radian, from the horizontal. Consequently, the vertical force is not wasted, however. It increases the load on the tires and thereby contributes to the braking force.

Reverse thrust loading is a parameter in the analysis. The horizontal component is specified as a fraction of aircraft gross weight, subject only to the constraint that the resultant reverse thrust not exceed the total engine thrust. Typical values are in the range 0.1 to 0.3.

Drag acts constantly on the moving airplane. Its magnitude is also a parameter in the analysis. The magnitude of the drag force at the initial velocity Vo is specified as a fraction of the aircraft gross weight. From this, a drag coefficient is computed and held constant for the duration of the integration.

In summary then, the landing scenario looks like this:

TIME	VELOCITY	EVENT
0	VO	Airplane touches down.

T1	V1	Brakes applied, throttle opened, thrust reverser deployed.
T2	V2	Full reverse thrust obtained.
T3	V3	Throttle closed, reverser stowed.
T4	0	Stopped.

2.3 Computer Program

ROLLOUT, the computer program developed for this analysis is listed in the Appendix. It is written in small modules, each of which does only one task. Thus, it is easy to change. For example, if one is interested in a particular configuration for which he has detailed drag data, he need change only subroutine AIRDRAG. Similarly, different engines may be simulated by changing the tabulated data in subroutine ENGINE. ROLLOUT accepts data in Namelist format as described in the Appendix. It then computes a set of 16 solutions, comprised of four reverse thrust loadings for each of four braking coefficients. Output consists of a printed table of distance versus time for each of the 16 solutions and three different sets of plots, two of which are optional. The first set of optional plots displays four curves of distance versus time corresponding to the four thrust loadings on one plot for each of the four braking coefficients. The other set of optional plots displays four sets of four small plots, each set corresponding to a braking coefficient. Each small plot shows the relative contribution of the brakes, the thrust reverser, and aerodynamic drag for the duration of the landing roll. The final plot is a crossplot of the 16 solutions, showing rolling distance versus braking coefficient for each of the four reverse thrust loadings.

2.4 Results

The results shown here represent a twin engine STOL transport weighing

160,000 pounds and having a landing speed of 150 feet per second or approximately 90 knots. Braking coefficients are as shown in Table 1 and the reverse thrust loadings were taken to be 0.1, 0.2, 0.3, and the maximum possible (about 0.316). The influences of the various parameters are discussed below.

A set of curves of rolling distance versus time (the first plot option) is shown in Figures 2, 4, 6, and 8. Figures 3, 5, 7, and 9 show the relative contributions of brakes, reverser, and drag for the same cases (the second option). Figure 10 contains the crossplot.

2.4.1 Braking Coefficient and Reverse Thrust Loading

One sees in Figures 2 through 9 the perhaps surprising result that, except on ice, the brakes do more to stop the airplane than does the thrust reverser. This is due in part at least to the particular reverser used here. The vertical component of the reversed thrust is 1.557 times the horizontal. By forcing the tires more firmly against the landing surface, it contributes to the braking force. A reverser which turned the exhaust jet more would of course alter the split.

Figure 10 shows that with the maximum reverse thrust it is possible to stop the airplane on ice in less than 1000 feet and on dry concrete in less than 600 feet. With the minimum reverse thrust, the braking force is still sufficient to stop the airplane in 900 feet on dry concrete, but on ice the distance required is much larger than the 1400 foot upper limit assumed (the actual distance was some 2100 feet). The utility of Figure 10 is that is shows the level of reverse thrust one must have to achieve a particular rolling distance. For instance, if the required maximum distance is 1000 feet for all surfaces, the reverse thrust loading must be at least 0.3.

2.4.2 **Engine Time Constant**

The effect of engine acceleration time is almost impossible to detect on plots of distance versus time. It is a little easier to see in plots of velocity versus time, as in Figure 11. Shown there are two velocity histories, one for a time constant yielding a half second acceleration. The principal difference is a displacement of about a quarter second (or about 9 ft/sec) and a more pronounced rounding of the initial part of the curve. The sharp bend at the right end of the curves shows the use of V_3 (here 10 ft/sec) to shut off the reversers. A cross plot of rolling distance versus time constant is shown in Figure 12.

2.4.3 Drag

For want of data describing the aerodynamic drag of the kind of aircraft considered here, we used the parametric approach. For example, DFR = 0.1 means that at the start of the calculation, $F_{dr}/GRWT = 0.1$. Defining $C_D = F_{dr}/V^2$ gives $C_D = DFR * GRWT/VO^2$. Figures 6 through 9 show that for DFR < 0.1, drag plays a very small part, while Figure 13 shows what can happen with very large drag forces.

3. CONCLUSIONS

The computer program described herein is a useful tool for quickly obtaining estimates of aircraft landing roll distances under various operating conditions. It permits independent evaluation of the various parameters which affect the rolling distance and can easily be modified to model configurations other than the twin engine STOL shown.

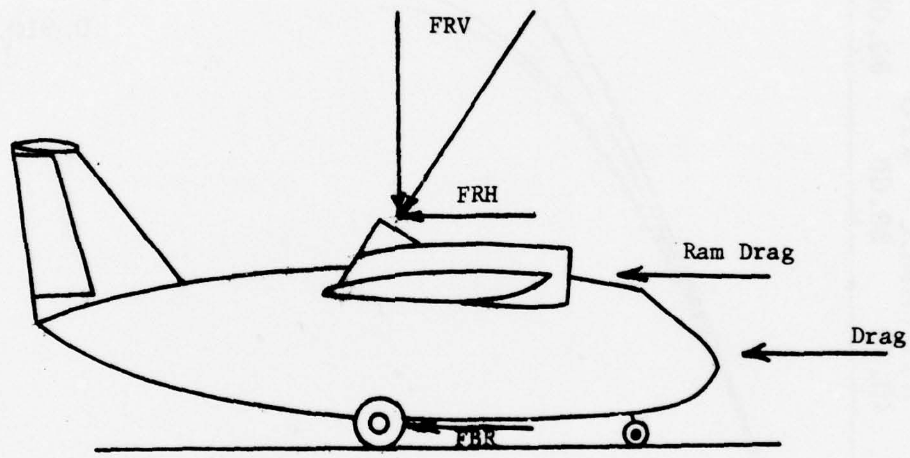


FIGURE 1

Forces on the Aircraft

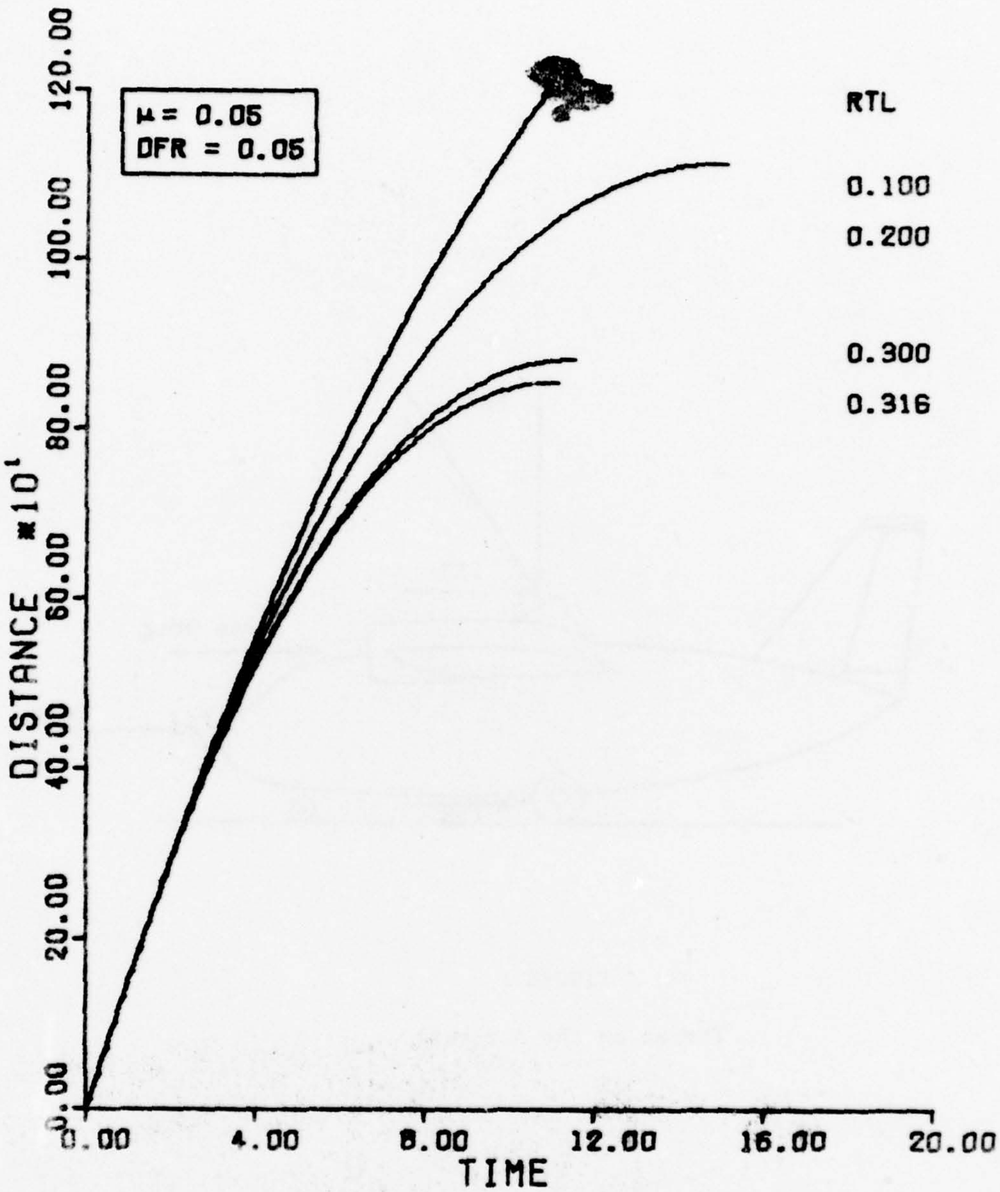


FIGURE 2

Rolling Distance on Ice

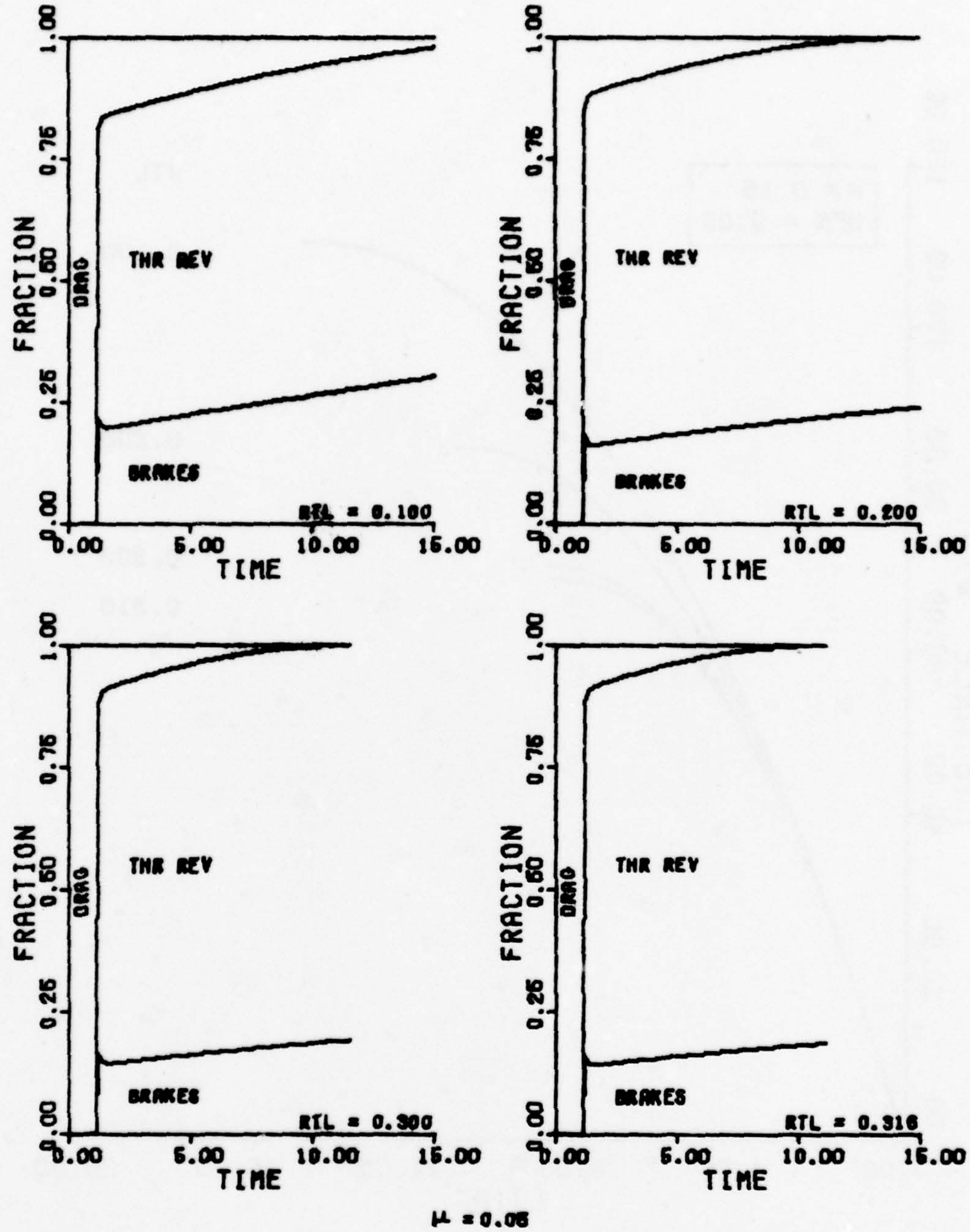


FIGURE 3

Relative Contributions of Drag, Brakes
and Thrust Reverser on Ice

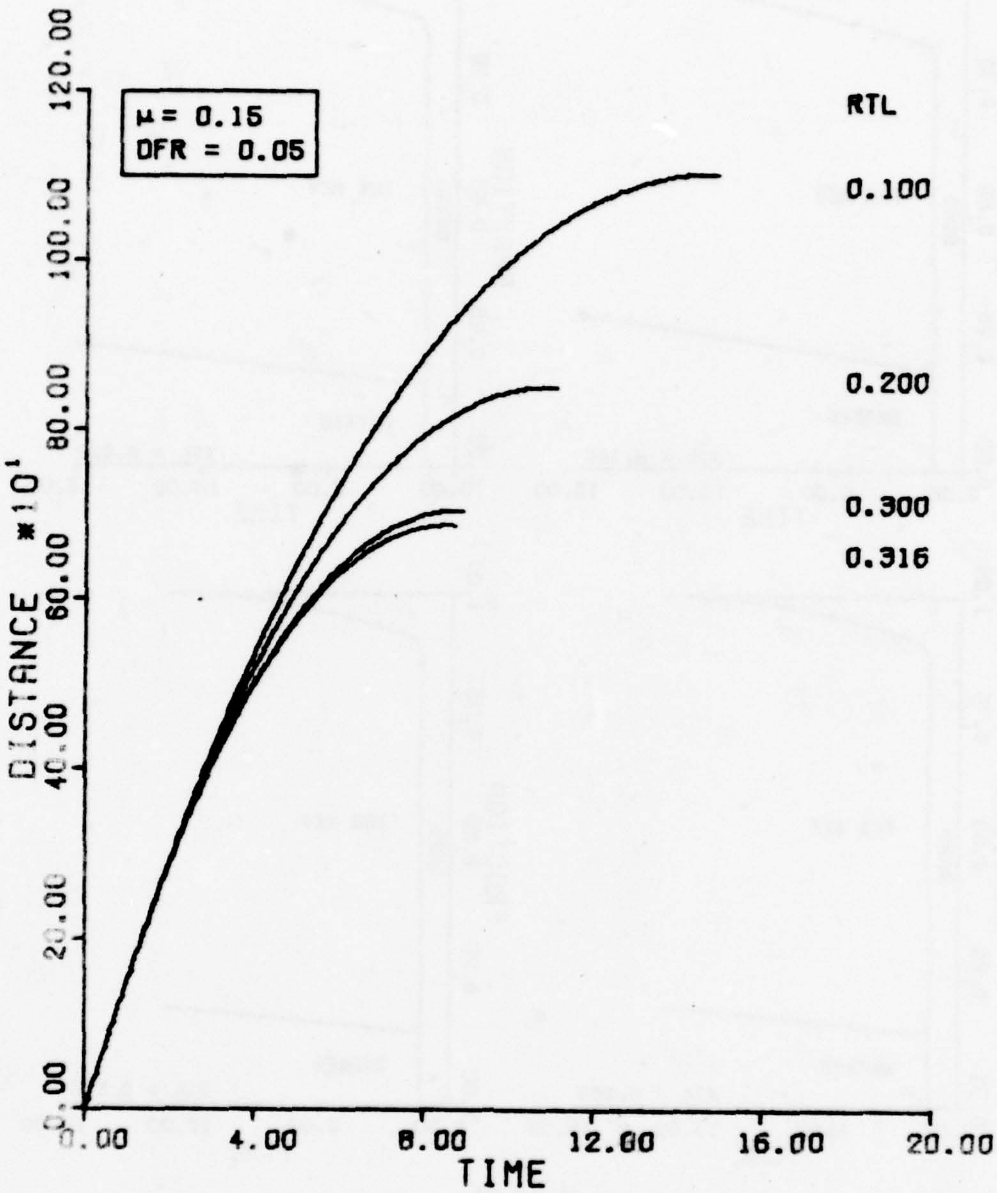


FIGURE 4

Rolling Distance on Wet Dirt

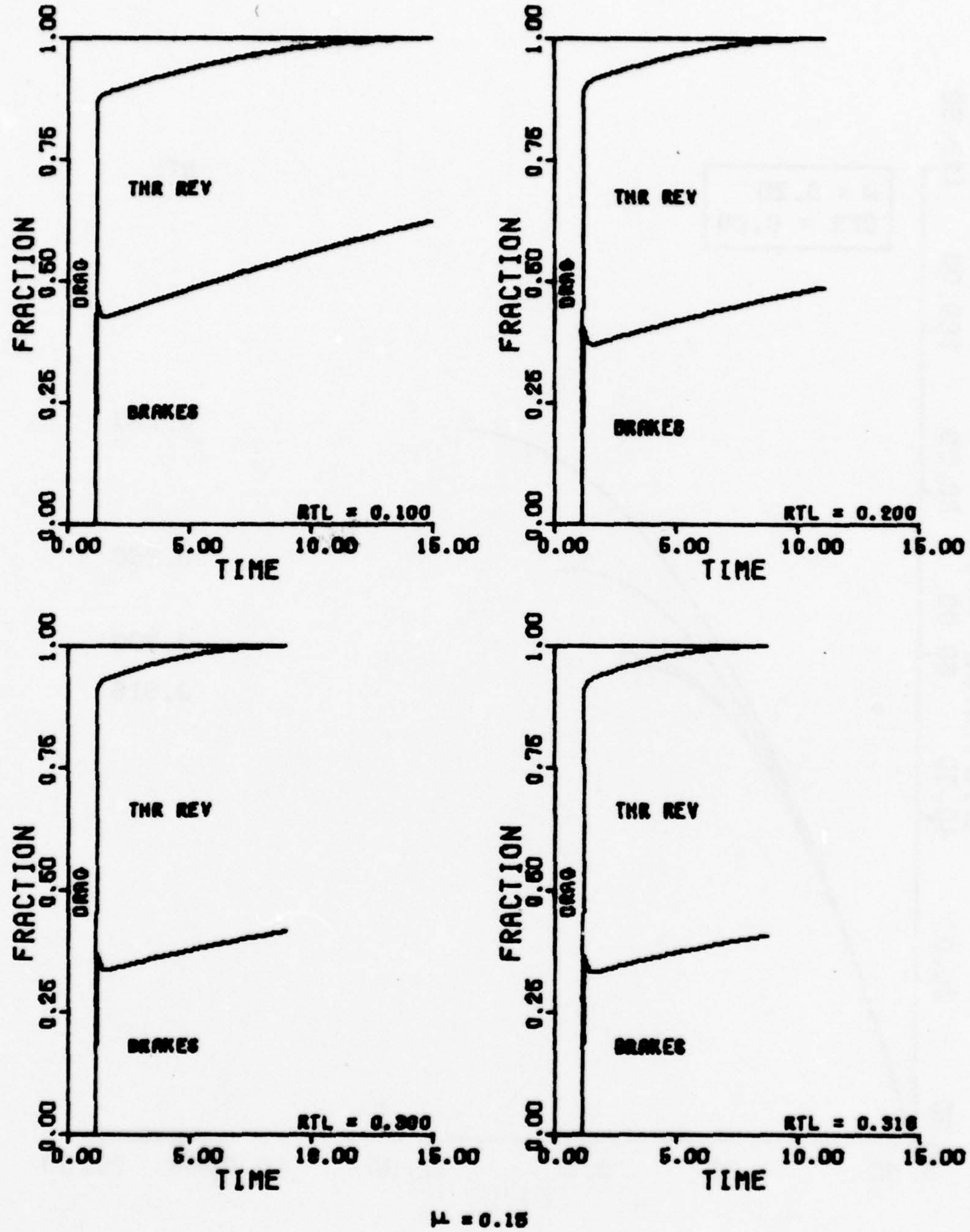


FIGURE 5

Relative Contributions of Drag, Brakes, and Thrust Reverser on Wet Dirt

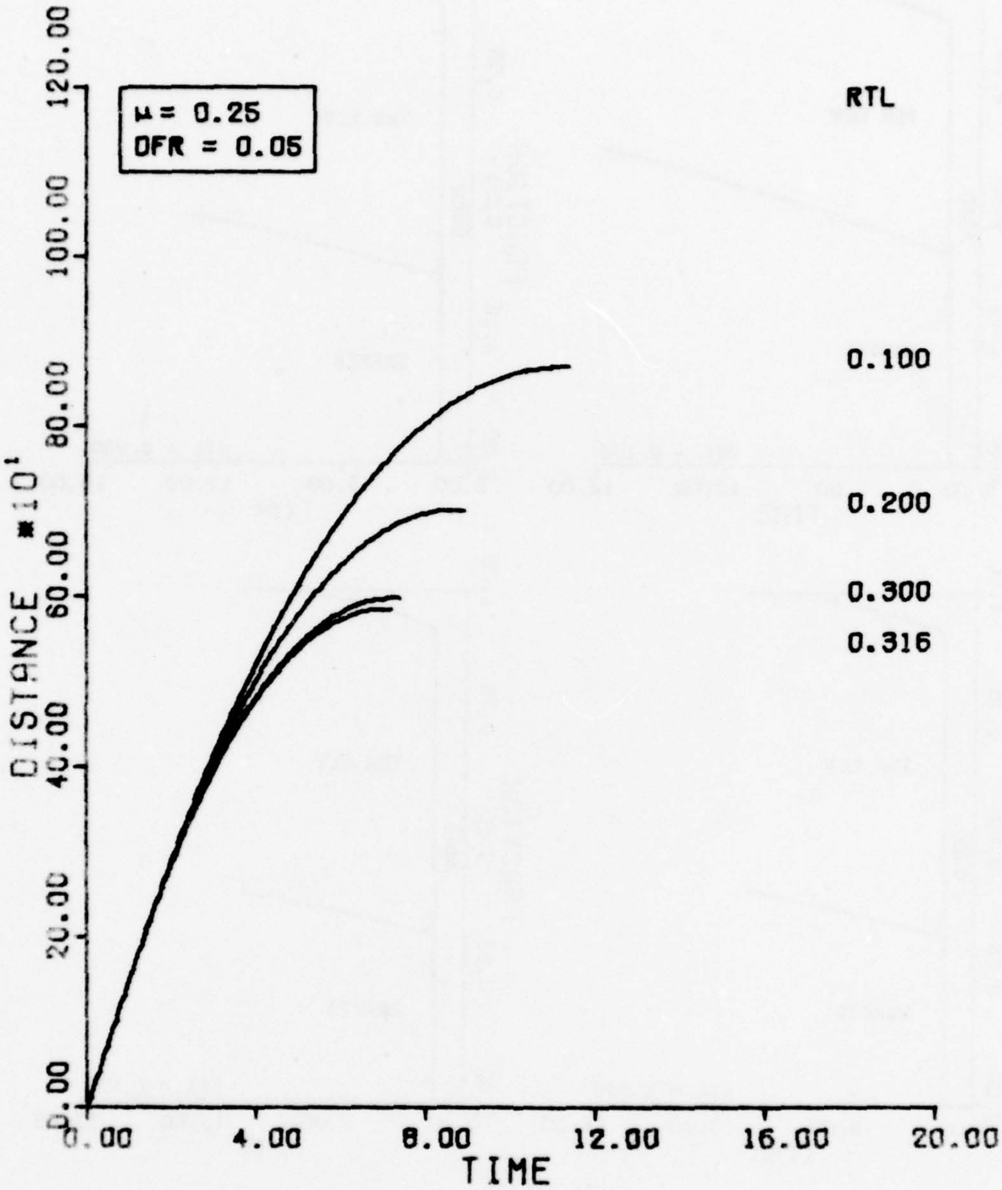


FIGURE 6

Rolling Distance on Dry Dirt

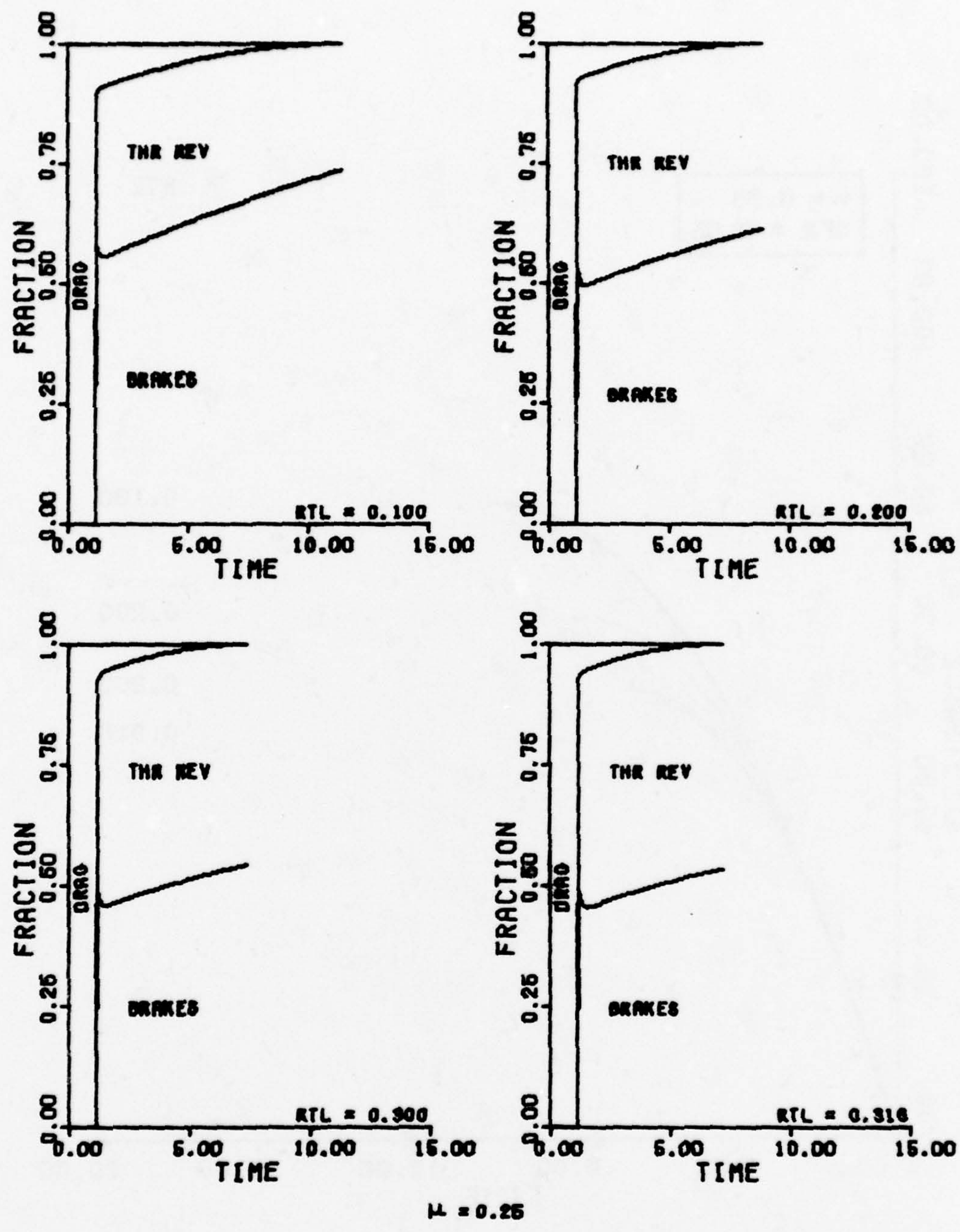


FIGURE 7

Relative Contributions of Drag, Brakes, and Thrust Reverser on Dry Dirt

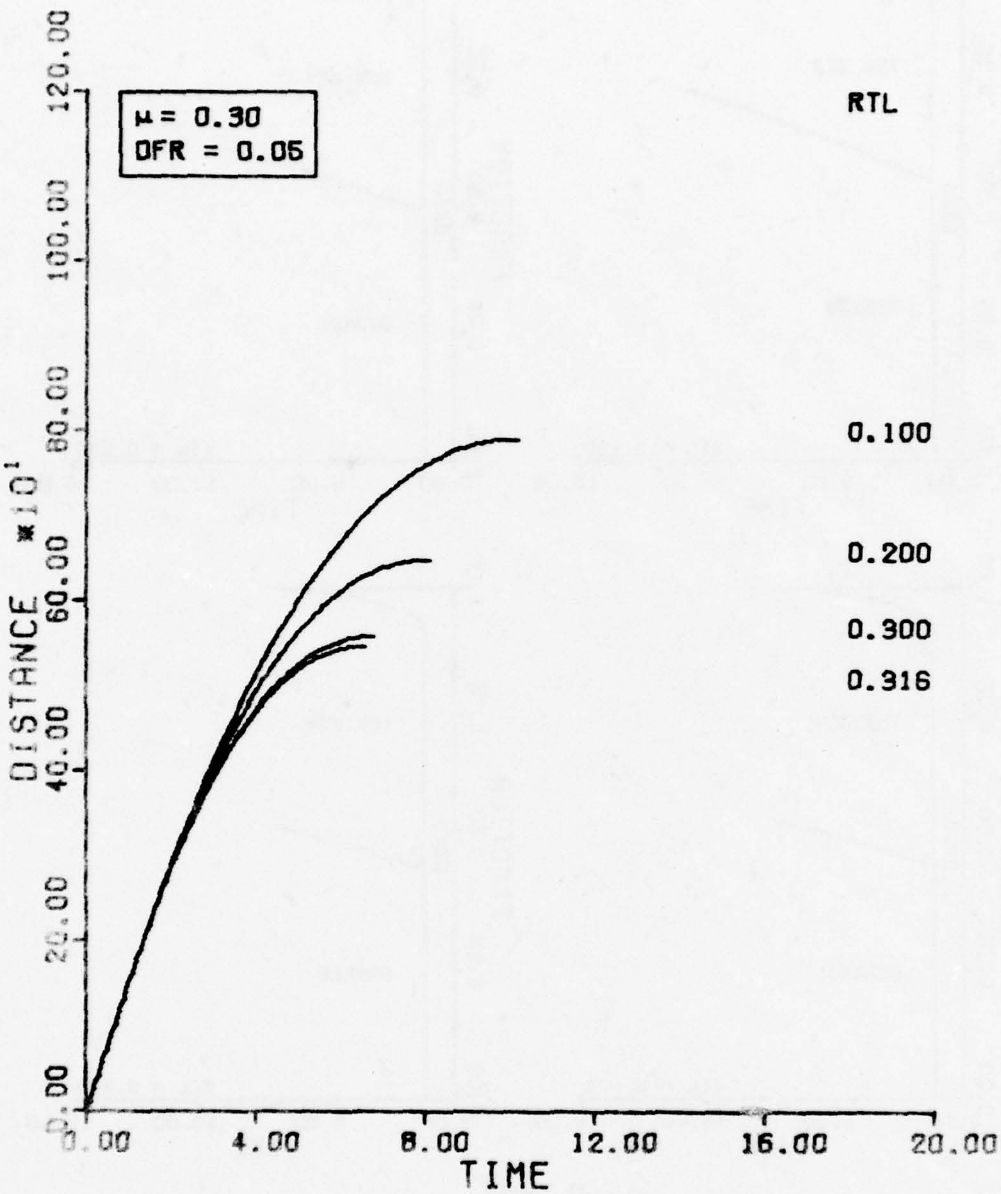


FIGURE 8

Rolling Distance on Dry Concrete

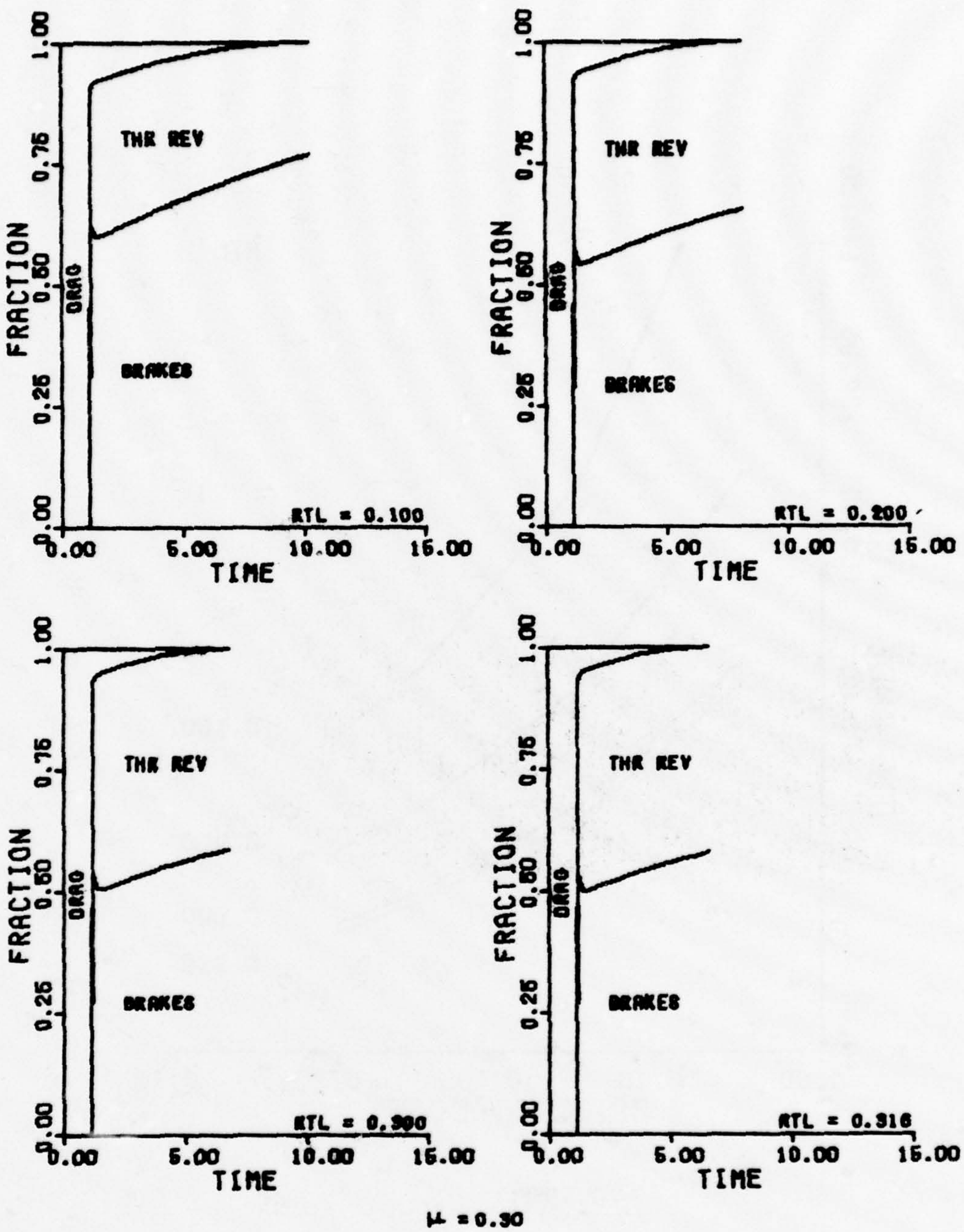


FIGURE 9

Relative Contributions of Drag, Brakes, and Thrust Reverser on Dry Concrete

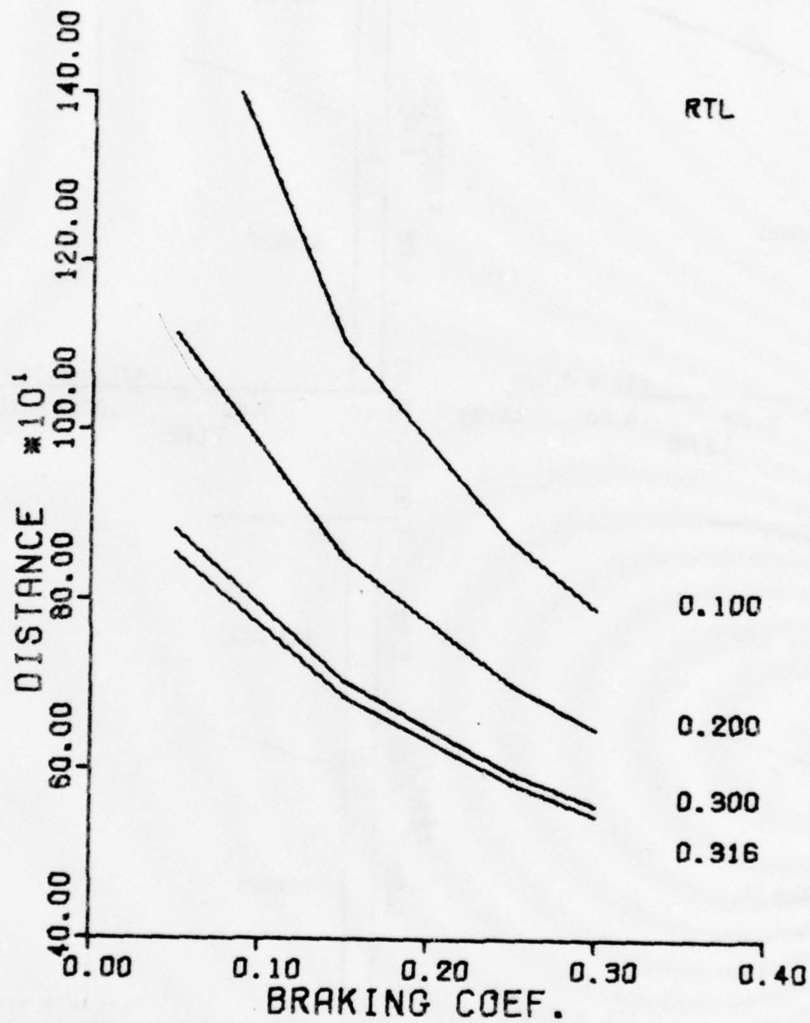


FIGURE 10

Crossplot

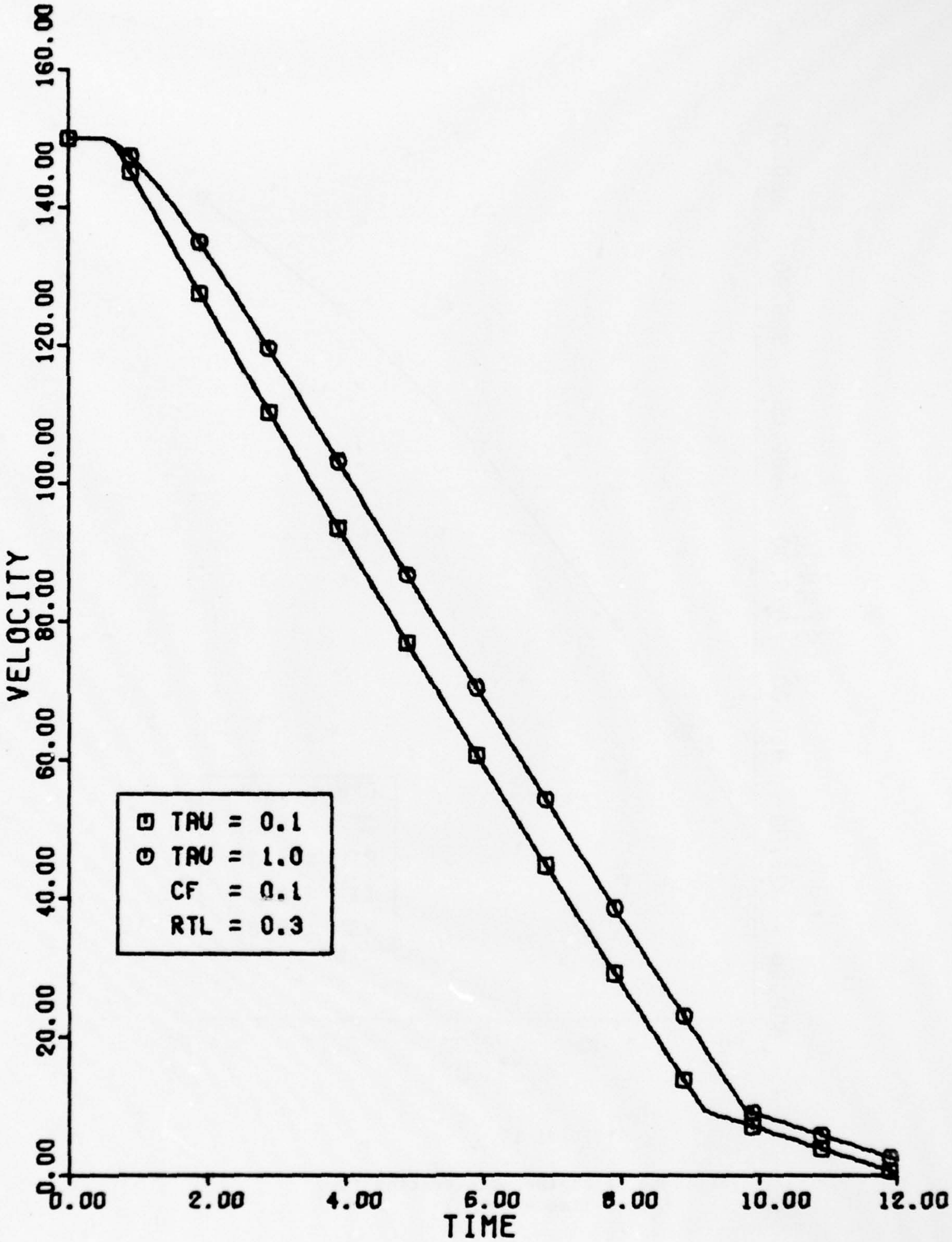


FIGURE 11

Velocity History

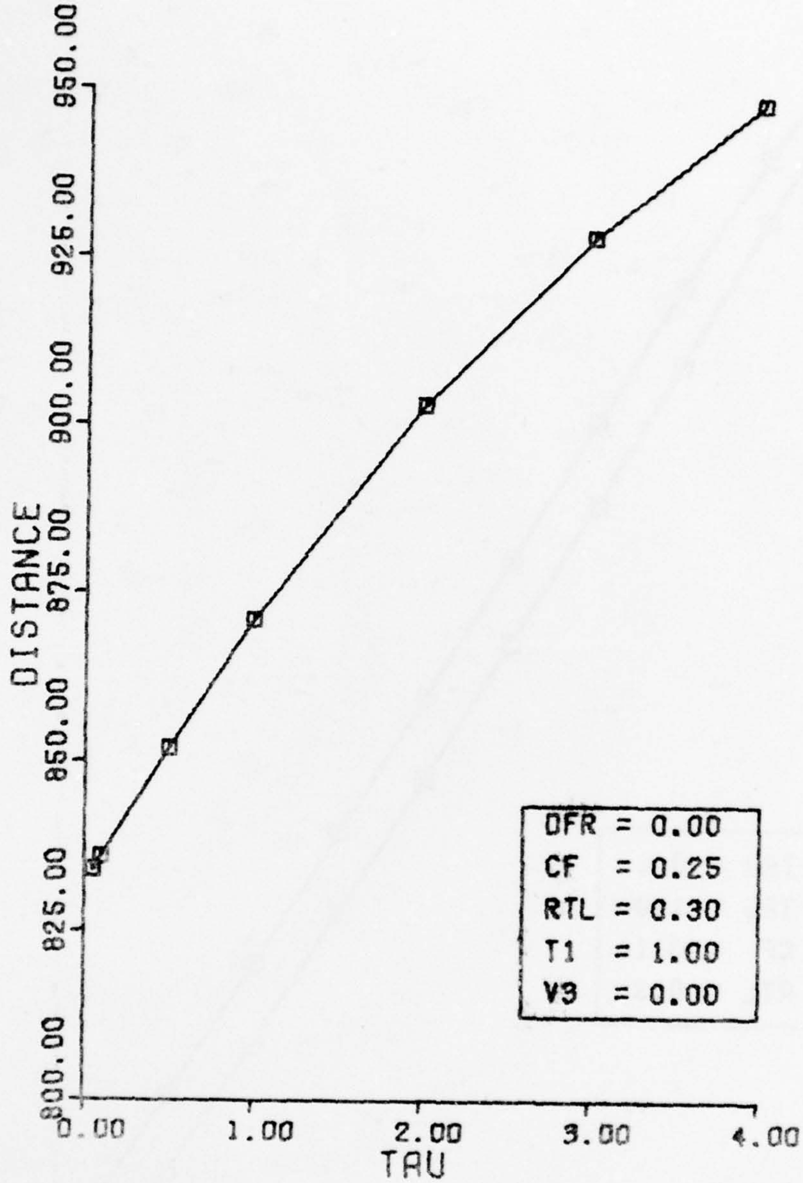


FIGURE 12

Influence of Engine Acceleration Time
(Accel Time = 5 TAU)

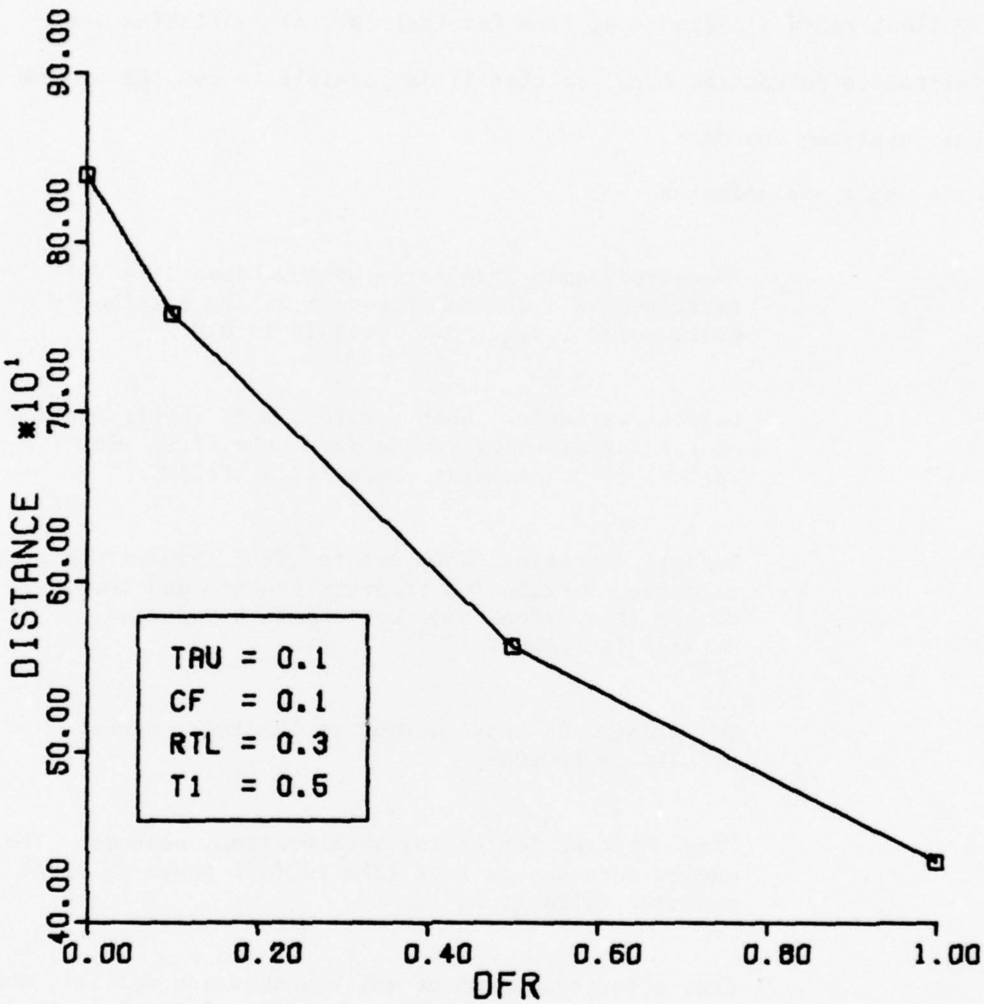


FIGURE 13

Influence of Drag

APPENDIX A

Input Instructions

Rollout reads all its input from Namelist IN. All variables are initialized in Subroutine INPUT so that it is possible to run the program without supplying any data.

The input variables are:

DFR	The aerodynamic drag force at touchdown (V = VO) expressed as a decimal fraction of the airplane's gross weight, e.g., 0.1 Default is 0.0.
DIS	Logical variable. When set to .TRUE. causes plots of rolling distance versus time (the first plot option) to be created. Default is .FALSE.
FRAC	Logical variable. When set to .TRUE. causes plots of relative contribution of drag, brakes, and thrust reverser (the second plot option) to be created. Default is .FALSE.
GRWT	The airplane's gross weight at landing, pounds. Default is 100,000.
TAU	Time constant for engine acceleration, seconds. The engine accelerates from idle to full power in 5/TAU seconds. Default is 1.
T1	Time after touchdown at which brakes are applied, thrust reverser opened, seconds. Typically 0.5. Default is 0.
VO	Airplane velocity at touchdown, feet per second. Default is 100. Setting VO negative stops the program.
V3	Velocity below which thrust reverser is no longer used, feet per second. Default is 20.

APPENDIX B

Program Listing

Program ROLLOUT and all its subroutines are listed on the following pages. ROLLOUT uses the Calcomp graphics subroutine library, though other graphics systems could easily be substituted. ROLLOUT was written in DCD Fortran IV Extended and contains a few dialect statement forms which are not compatible with other compilers. All are easily changed, however. There are a few statements of the form A = B = C = 0.0 and some sever character variable names. The forms READ * and PRINT * are list-directed (format-free) input-output.

The listings contain occasional continuation lines marked with a dollar sign. These were created by the listing program to maintain the right-hand margin and do not appear in the actual code.

```
      PROGRAM  ROLLOUT (INPUT=101, OUTPUT, TAPE6=OUTPUT,
     $ PLOT)
C
      COMMON  /GIVEN/  V0, V3, GRWT, THRMAX, CF, T1, T2,
     $ T3, TAU, CVR
     1  (2), RTC, RNO, DFR
      COMMON  /DCC/  DC
      COMMON  /RETARD/  IRT, RRF (250,4), DRF (250,4), RVF
     $ (250,4)
      COMMON  /PLOTFL/ DIS, FRAC
      LOGICAL      DIS, FRAC
C
C        CFF ARE THE BRAKING COEFFICIENTS FOR ICE, WET &
C    $          DRY
C        GRASS, AND DRY CONCRETE.
C        RTL IS THE HORIZONTAL COMPONENT OF REVERSE THRUST
C        DIVIDED BY VEHICLE GROSS WEIGHT.  THERE IS ALSO A
C        VERTICAL COMPONENT.  THE RESULTANT REVERSE THRUST
C        VECTOR IS INCLINED AT 57.3 DEG TO THE HORIZONTAL.
C
      DIMENSION   CFF (4), D (4,4), NTG (4), RTL (4), T
     $ (250), V (250),
     1   X (250), XP (250,4)
      DATA      CFF / 0.05, 0.15, 0.25, 0.30 /
      DATA      RTL / 0.1, 0.2, 0.3, 0.35 /
C
      RNO          = 0.0
      CALL PLOT (1.0, 1.0,  - 3)
    1 CALL INPUT
      IF (V0 .LE. 0.0) GO TO 4
C
C        DFR IS THE DRAG FORCE AT V = V0 AS A FRACTION
C        OF GROSS WEIGHT.
C        DC IS THE EQUIVALENT DRAG COEFFICIENT.
C
      DC           = DFR * GRWT / (V0 * V0)
C
        DO 3 ICF  = 1, 4
        CF          = CFF(ICF)
          DO 2 IRT  = 1, 4
C
C         RUN FOUR REVERSE THRUST LOADINGS FOR EACH
C         BRAKING COEFFICIENT.
C
          CALL CFR (RTL(IRT))
          RTC          = RTL(IRT)
          CALL COMPUTE (X, V, T, 250, NINTG)
          RNO          = RNO + 1.0
          CALL OUTPUT (X, V, T, 250, NINTG)
```

```
            NTG(IRT)    = NINTG
            D(ICF, IRT) = X(NINTG)
C
              DO 2 I     = 1, NINTG
              XP(I, IRT) = X(I)
    2         CONTINUE
          IF (DIS) CALL GRAPHIC (T, XP, NTG, RTL, 250, 4)
          IF (FRAC) CALL QUAD (CF, NTG, RTL, T, 4, 250)
    3     CONTINUE
        CALL CROSFLT (D, CFF, RTL, 4)
        GO TO 1
    4   CALL SYMBOL (0.0, 0.5, 0.105, "FINISHED", 90.0, 8)
        CALL PLOTE
        STOP
        END
```

```
      SUBROUTINE  AIRDRAG (V, DRAG)
      COMMON  /DCC/  DC
C
      DRAG        = DC * V * V
      RETURN
      END
```

```
      SUBROUTINE  BRAKES (DOWNF, T1, T, CF, FBR)
C
C        COMPUTES BRAKING FORCE
C
      IF (T .LT. T1) GO TO 1
      FBR        = CF * DOWNF
      RETURN
    1 FBR        = 0.0
C
C        NO BRAKES FOR T < T1
C
      RETURN
      END
```

```
      SUBROUTINE  CFF (RTL)
      COMMON  /THRUST/  FGRD (5)
C
C        COMPUTES REV. THR. COEFF. FROM REV. THR. LOADING
C
      COMMON  /GIVEN/  V0, V3, GRWT, THRMAX, CF, T1, T2,
     $ T3, TAU, CVR
     1  (2), RTC, RNC, DFF
      CVR(1)     = RTL * GRWT / (1.92 * FGRD(1))
      CVR(2)     = 1.5577 * CVR(1)
C
C        1.5577 = TAN(57.3 DEG.)
C        1.92   = 2*0.96
C
C        GUARD AGAINST RESULTANT CV > 1.
C
      IF (CVR(1) .LE. 0.5403) RETURN
C
C        THESE ARE THE COSINE AND SINE OF 57.3 DEG.
C
      CVR(1)     = 0.5403
      CVR(2)     = 0.8415
      RTL        = 1.0374 * FGRD(1) / GRWT
C
C        1.0374 = 1.92*0.5403
C
      PRINT *, "RTL HAD TO BE REDUCED TO ", RTL
      RETURN
      END
```

```
      SUBROUTINE  COMPUTE (X, V, T, NO, NINTG)
      COMMON  /GIVEN/  V0, V3, GRWT, THRMAX, CF, T1, T2,
     $ T3, TAU, CVR
     1  (2), RTC, RNO, DFP
      COMMON  /RETARD/  IRT, BRF (250,4), DRF (250,4), RVF
     $ (250,4)
C
C        INTEGRATES THE EQUATION OF MOTION, F = MA, IN A
C        TAYLOR SERIES EXPANSION, VIZ
C        X(T+DT) = X(T) + DT*(DX/DT) + (DT**2/2)*(D2X/DT2)
C        WHERE DX/DT = V
C              D2X/DT2 = A = F/M
C              F = DRAG + BRAKING + REV. THRUST
C
C        BEGIN
C
      DIMENSION    T (NO), V (NO), X (NO)
      DT           = 0.1
      DT2          = DT * DT / 2.0
      X(1)         = 0.0
      V(1)         = V0
      T(1)         = 0.0
      OM           = 32.174 / GRWT
      T2           = 3600.0
      T3           = 3600.0
      NMAX         = NO - 2
      BRF(1, IRT) = 0.0
      DRF(1, IRT) = 0.0
      RVF(1, IRT) = 0.0
      IF (DFP .GT. 0.0) DRF(1, IRT) = 1.0
C
C        INTEGRATION LOOP
C
        DO 3 I     = 2, NMAX
        CALL AIRDRAG (V(I - 1), DRAG)
        CALL ENGINE (V(I - 1), THRMAX, RAMDRAG)
        CALL SECOLUF (T(I - 1), T1, T2, T3, TAU, THRMAX,
     $ THR)
        CALL REVTHR (THR, CVR, V3, V(I - 1), FRH, FRV)
        CALL BRAKES (GRWT + FRV, T1, T(I - 1), CF, FBR)
        IF (FRH .GT. 0.0) FRH = FRH + RAMDRAG
        FSUM        = DRAG + FBR + FRH
        IF (FSUM .GT. 0.0) GO TO 1
        BRF(I, IRT) = BRF(1, IRT)
        DRF(I, IRT) = DRF(1, IRT)
        RVF(I, IRT) = RVF(1, IRT)
        GO TO 2
    1   BRF(I, IRT) = FBR / FSUM
        DRF(I, IRT) = DRAG / FSUM
```

```
      RVF(I, IRT) = FRH / FSUM
 2    XPP         = - OM * FSUM
      V(I)        = V(I - 1) + DT * XPP
      X(I)        = X(I - 1) + DT * V(I) + DT2 * XPP
      T(I)        = T(I - 1) + DT
C
C      WHEN V DROPS BELOW V3, TURN OFF THRUST REVERSERS.
C
      IF  (V(I) .LE. V3 .AND. T3 .GE. 3600.0) T3 = T (I)
C
C      ARE WE STOPPED?
C
      IF (V(I) .LE. 0.05) GO TO 4
 3    CONTINUE
    I           = NMAX
C
C      NINTG IS THE NUMBER OF INTEGRATION STEPS.
C
 4  NINTG       = I
    IF (T3 .GE. 3600.0) T3 = T(I)
    RETURN
    END
```

```
      SUBROUTINE  CROSPLT (DIST, CFF, RTL, N)
      COMMON  /FRAME/  XF, YF
      DIMENSION   BC (6), CFF (N), DIST (N,N), RTL (N) , X
     $ (6)
      DATA      YNUM / 6.0 /
C
C         PLOTS LANDING DISTANCE VS BRAKING COEFFICIENT
C         WITH REVERSE THRUST LOADING AS A PARAMETER.
C
      DATA      X (5), X (6) / 400.0, 200.0 /
      DATA      BC (5), BC (6) / 0.0, 0.1 /
      XF           = 4.0
      YF           = 5.0
      CALL PLOT (2.0, 2.0, -3)
      CALL AXIS (0.0, 0.0, 13HBRAKING COFF.,  - 13, XF,
     $ 0.0, BC(5),
     1   BC(6))
      CALL AXIS (0.0, 0.0, 8HDISTANCE, 8, YF, 90.0, X(5),
     $ X(6))
      CALL SYMBOL (3.5, 4.9, 0.1, 3HRTL, 0.0, 3)
      YO           = YNUM
C
        DO 1 I      = 1, 4
        BC(I)       = CFF(I)
    1   CONTINUE
C
        DO 3 I      = 1, N
          DO 2 J      = 1, 4
          X(J)        = DIST(J, I)
    2     CONTINUE
        CALL CLIP (BC, X, N, 1, 0, 0)
C
C          Y LOCATION TO DRAW REVERSE THRUST LOADING VALUE
C
        YN           = (X(4) - 400.0) / 200.0
C
C          MAKE SURE THEY DON'T OVERLAP
C
        IF  ((YO - YN) .LT. 0.3) YN = YO - 0.3
        YO           = YN
        CALL NUMBER (3.5, YN, 0.1, RTL(I), 0.0,  3)
    3   CONTINUE
      CALL PLOT (8.0, -2.0, -3)
      RETURN
      END
```

```
      SUBROUTINE  ENGINE (V, THRMAX, RAMDRAG)
      COMMON  /THRUST/  FGSL (5)
      DIMENSION   RDSL (5), VSL (5)
      DATA     VSL / 0.0, 55.82, 111.64, 167.46, 223.28 /
      DATA     FGSL / 48700.0, 49000.0, 49300.0, 50000.0,
     $ 50900.0 /
      DATA     RDSL / 0.0, 2600.0, 5100.0, 7700.0, 10700.0
     $ /
C
C        DATA FOR ONE ENGINE AT SEA LEVEL.
C        FGSL IS GROSS THRUST (CV = 1.0).
C        RDSL IS THE RAM DRAG.
C
C        FIND V IN VSL TABLE
C
      IF (V .GT. VSL(3)) GO TO 2
      IF (V .GT. VSL(2)) GO TO 1
      I          = 1
      GO TO 4
    1 I          = 2
      GO TO 4
    2 IF (V .GT. VSL(4)) GO TO 3
      I          = 3
      GO TO 4
    3 I          = 4
C
C        ASSUME TWO ENGINES
C
    4 Z          = (V - VSL(I)) / (VSL(I + 1) - VSL(I))
      THRMAX     = 2.0 * (FGSL(I) + Z * (FGSL(I + 1) -
     $ FGSL(I)))
      RAMDRAG    = 2.0 * (RDSL(I) + Z * (RDSL(I + 1) -
     $ RDSL(I)))
      RETURN
      END
```

```
      SUBROUTINE  GRAPHIC (T, X, NTG, RTL, NO, MO)
      COMMON  /GIVEN/  V0, V3, GPWT, THPMAX, CF, T1, T2,
     $ T3, TAU, CVR
     1  (2), RTC, RNO, DFR
      COMMON  /FRAME/  XF, YF
      DIMENSION   NTG (MO), RTL (MO), T (NO), X (NO,MO)
      DATA      YNUM / 5.7 /
C
C        ON LINE CALCOMP
C
C        DRAW AXES AND PLOT X VS. T
C
      XF          = 5.0
      YF          = 6.0
      CALL PLOT  (0.0, 1.5, -3)
      CALL AXIS  (0.0, 0.0, 4HTIME,  - 4, XF, 0.0, 0.0,
     $ 4.0)
      CALL AXIS  (0.0, 0.0, 8HDISTANCE, 8, YF, 90.0, 0.0,
     $ 200.0)
C
C        ANNOTATE
C
      CALL SYMBOL (0.30, 5.8, 0.14, 102, 0.0, -1)
      CALL SYMBOL (0.45, 5.8, 0.10, 1H=, 0.0, 1)
      CALL NUMBER (0.65, 5.8, 0.10, CF, 0.0, 2)
      CALL SYMBOL (0.30, 5.6, 0.10, 5HDFR =, 0.0, 5)
      CALL NUMBER (0.90, 5.6, 0.10, DFR, 0.0, 2)
C
C        DRAW BOX
C
      CALL PLOT  (0.20, 5.5, 3)
      CALL PLOT  (0.20, 6.0, 2)
      CALL PLOT  (1.35, 6.0, 2)
      CALL PLOT  (1.35, 5.5, 2)
      CALL PLOT  (0.20, 5.5, 2)
C
      CALL SYMBOL (4.5, 5.9, 0.1, 3HRTL, 0.0, 3)
      Y0          = YNUM
C
        DO 1 I    = 1, MO
        NM        = NTG(I)  + 1
        NS        = NM + 1
C
C        INSERT SCALE FACTORS INTO T ARRAY.
C
        TNM       = T(NM)
        TNS       = T(NS)
        T(NM)     = 0.0
        T(NS)     = 4.0
```

```
          X(NM, I)   = 0.0
          X(NS, I)   = 200.0
          CALL CLIP (T, X(1,I), NTG(I), 1, 0, 0)
          T(NM)      = TNM
          T(NS)      = TNS
C
C          YN IS THE VERTICAL LOCATION OF THE RTL LABEL.
C
          YN         = X(NTG(I), I) / 200.0
          IF   ((YO - YN) .LT. 0.3) YN = YO - 0.3
          YO         = YN
          CALL NUMBER (4.5, YN, 0.1, RTL(I), 0.0, 3)
    1     CONTINUE
       CALL PLOT (8.5, -1.5, -3)
       RETURN
       END
```

```
      SUBROUTINE  INPUT
      COMMON  /GIVEN/  V0, V3, GRWT, THRMAX, CF, T1, T2,
    $ T3, TAU, CVR
    1  (2), RTC, RNC, DFR
      COMMON  /PLCTRL/ DIS, FRAC
      LOGICAL      DIS, FRAC
      DATA     V0, V3, GRWT, THRMAX, CF, T1, T3, TAU, CVR /
    $ 100.0, 20.0,
    1   1.0E05, 5.0E04, 0.2, 0.5, 0.0, 1.0, 0.5, 0.1 /
      DATA     DFR / 0.0 /
      DATA     DIS, FRAC / 2 * .FALSE. /
      NAMELIST  / IN / DFR, DIS, FRAC, GRWT, TAU, T1, V0,
    $ V3
      READ  IN
      RETURN
      END
```

```
      SUBROUTINE  OUTPUT (X, V, T, NO, NINTG)
      COMMON  /GIVEN/  V0, V3, GRWT, THRMAX, CF, T1, T2,
     $ T3, TAU, CVR
     1  (2), RTC, RNO, DFR
C
C          PRINT HEADER
C
      DIMENSION   T (NO), V (NO), X (NO)
      PRINT  100, GRWT, THRMAX, CVR, RTC, DFR, T1, T2, T3,
     $ V3, V0,
     1   CF, TAU, RNO
C
C          PRINT T, V, X ARRAYS
C
        DO 2 K     = 1, 3
        PRINT  101
        I1          = 1 + 100 * (K - 1)
        I2          = I1 + 49
        I2          = MIN0(I2, NO - 2, NINTG)
          DO 1 I    = I1, I2
          PRINT  102, T (I), V (I), X (I)
          IF (I+50 .LE. NINTG) PRINT  103,T (I + 50),V (I
     $ + 50),X (I +
     1        50)
    1     CONTINUE
        IF (I2 .GE. NINTG .OR. I1+100 .GT. NINTG) RETURN
        PRINT  104
    2   CONTINUE
      RETURN
C
  100 FORMAT  ( 1H1, 3X,  *GR.WT. =* F8.0,  * MAX THRUST
     $ =* F8.0,
     1   * CVRH =* F5.3, * CVRV =*, F5.3, * RTL =* F5.3,
     $   * DFR =* F5.3 / 4X, *T1 =*
     2   F6.2,  * T2 =* F6.2,  * T3 =* F6.2,  * V3 =* F7.0,
     $   * V0 =*
     3   F7.0,  * CF =* F5.3,  * TAU =* F5.3,  * RUN NO. =*
     $ F3.0 )
  101 FORMAT  ( 1H0, 2(6X, 4HTIME, 2X, 8HVELOCITY, 2X,
     $ 8HDISTANCE, 10X)
     1   )
  102 FORMAT  ( 1H , 3F10.2 )
  103 FORMAT  ( 1H+, 40X, 3F10.2 )
  104 FORMAT  ( 1H1  )
C
      END
```

```
      SUBROUTINE  QUAD (CF, NTG, RTL, T, N, NT)
      COMMON  /RETARD/  IRT, BRF (250,4), DRF (250,4), RVF
     $ (250,4)
      COMMON  /FRAME/  XF, YF
      DIMENSION   NTG (N), RTL (N), T (NT), XO (4), YO (4)
C
C        PLOTS A GRAPH FOR EACH OF FOUR THRUST LOADINGS.
C        EACH GRAPH SHOWS THE RELATIVE CONTRIBUTION OF
C    $         DRAG,
C        BRAKES, AND THRUST REVERSER.
C
      DATA     XO / 2.0, 4.0, -4.0, 4.0 /
      DATA     YO / 5.0, 0.0, -5.0, 0.0 /
      DATA     F1, F2 / 0.75, 1.0 /
      YF          = 4.0
      XF          = 3.0
      CALL PLOT (1.0, 1.5, -3)
      CALL FACTOR (F1)
        DO 1 L     = 1, 4
        CALL PLOT (XO(L), YO(L), -3)
        NL         = NTG(L)
          DO 2 I     = 1, NL
          RVF(I, L) = RVF(I, L) + BRF(I, L)
          DRF(I, L) = DRF(I, L) + RVF(I, L)
    2     CONTINUE
        NM         = NTG(L) + 1
        NS         = NM + 1
        TNM        = T(NM)
        TNS        = T(NS)
        T(NM)      = 0.0
        T(NS)      = 5.0
        BRF(NM, L) = DRF(NM, L) = RVF(NM, L) = 0.0
        BRF(NS, L) = DRF(NS, L) = RVF(NS, L) = 0.25
        CALL AXIS (0.0, 0.0, 4HTIME, -4, XF, 0.0, 0.0,
     $ 5.0)
        CALL AXIS (0.0, 0.0, 8HFRACTION, 8, YF, 90.0, 0.0,
     $ 0.25)
        CALL CLIP (T, BRF(1, L), NTG(L), 1, 0, 0)
        CALL CLIP (T, RVF(1, L), NTG(L), 1, 0, 0)
        CALL CLIP (T, DRF(1, L), NTG(L), 1, 0, 0)
        T(NM)      = TNM
        T(NS)      = TNS
C
C        ANNOTATIONS
C
C        THRUST LOADING
C
        CALL SYMBOL (1.9, 0.05, 0.10, 5HRTL =, 0.0, 5)
        CALL NUMBER (2.5, 0.05, 0.10, RTL(L), 0.0, 3)
```

```
C
      YBLB      = 2.0 * BRF(38, L) - 0.05
      YTLB      = 2.0 * (BRF(38, L) + RVF(38, L)) - 0.05
      CALL SYMBOL (0.50, YBLB, 0.10, 6HBRAKES, 0.0, 6)
      CALL SYMBOL (0.50, YTLB, 0.10, 7HTHR REV, 0.0, 7)
      CALL SYMBOL (0.15, 1.80, 0.10, 4HDRAG, 90.0, 4)
    1 CONTINUE
C
C          BRAKING COEFFICIENT
C
      CALL SYMBOL (-1.00, -0.75, 0.20, 102, 0.0, -1)
      CALL SYMBOL (-0.75, -0.75, 0.10, 1H=, 0.0, 1)
      CALL NUMBER (-0.60, -0.75, 0.10, CF, 0.0, 2)
      CALL FACTOR (F2)
      CALL PLOT (5.0, -1.5, -3)
      RETURN
      END
```

```
      SUBROUTINE  REVTHR (THR, CVR, V3, V, FRH, FRV)
C
C        COMPUTES HORIZONTAL AND VERTICAL COMPONENTS
C        OF REVERSE THRUST.
C
      DIMENSION   CVR (2)
      IF (V .LE. V3) GO TO 1
      FRH        = THR * CVR(1)
      FRV        = THR * CVR(2)
      RETURN
C
C        NO REVERSE THRUST WHEN V < V3
C
    1 FRH        = FRV = 0.0
      RETURN
      END
```

```
      SUBROUTINE  SECOLUP (T, T1, T2, T3, TAU, THRMAX,
     $ THR)
C
C        COMPUTES ENGINE THRUST BETWEEN IDLE AND FULL
C        USING A FIRST ORDER LAG.
C
      IF (T .LT. T1 .OR. T .GT. T3) GO TO 1
      THR         = THRMAX * (1.0 - EXP( - (T - T1) / TAU))
C
C        FULL POWER IS REACHED AT T = T2.
C
      IF  ((T - T1) / TAU .GE. 5.0 .AND. T2 .GE. 3600.0)
     $ T2 = T
      RETURN
C
C        THROTTLE IS CLOSED FOR T < T1 OR T > T3
C
    1 THR         = 0.0
      RETURN
      END
```

```
      SUBROUTINE  CLIP (X, Y, N, K, J, L)
      COMMON  /ZSZYZN/  S1, S2, LS
      LOGICAL        S1, S2
      DIMENSION      X (1), Y (1)
      XIN(X)      = (X - XMIN) / DX
      YIN(Y)      = (Y - YMIN) / DY
      XMIN        = X(N + 1)
      YMIN        = Y(N + 1)
      DX          = X(N + 2)
      DY          = Y(N + 2)
      X1          = XIN(X(1))
      Y1          = YIN(Y(1))
      S1          = J .GT. 0
      S2          = .FALSE.
      LS          = L
        DO 1 I     = 2, N
        X2         = XIN(X(I))
        Y2         = YIN(Y(I))
        IF  (J .GT. 0) S2 = MOD(I, J)  .EQ. 0
        CALL AGAINST (X1, Y1, X2, Y2)
        X1         = X2
        Y1         = Y2
        S1         = .FALSE.
        S2         = .FALSE.
1       CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  AGAINST (X1, Y1, X2, Y2)
C
C        CLIPS THE LINE FROM (X1,Y1) TO (X2,Y2) AGAINST
C        0 <= X <= XF, 0 <= Y <= YF
C
      COMMON  /FRAME/  XF, YF
      COMMON  /ZSZYZW/  S1, S2, LS
      LOGICAL       S1, S2, TS
C
      K2          = KODE(X2, Y2)
    1 K1          = KODE (X1, Y1)
      IF (K1 .EQ. 0 .AND. K2 .EQ. 0) GO TO 6
      IF ((K1 .AND. K2) .NE. 0) RETURN
      IF (K1 .NE. 0) GO TO 2
C
C        SWAP ENDS
C
      T           = X1
      X1          = X2
      X2          = T
      T           = Y1
      Y1          = Y2
      Y2          = T
      K           = K1
      K1          = K2
      K2          = K
      TS          = S1
      S1          = S2
      S2          = TS
    2 IF ((K1 .AND. 1) .EQ. 0) GO TO 3
C
C        LEFT SIDE (X = 0)
C
      Y1          = Y1 - X1 * (Y2 - Y1) / (X2 - X1)
      X1          = 0.0
      GO TO 1
    3 IF ((K1 .AND. 2) .EQ. 0) GO TO 4
C
C        RIGHT SIDE (X = XF)
C
      Y1          = Y1 + (XF - X1) * (Y2 - Y1) / (X2 - X1)
      X1          = XF
      GO TO 1
    4 IF ((K1 .AND. 4) .EQ. 0) GO TO 5
C
C        BOTTOM (Y = 0)
C
      X1          = X1 - Y1 * (X2 - X1) / (Y2 - Y1)
      Y1          = 0.0
```

```
      GO TO 1
    5 IF ((K1 .AND. 8) .EQ. 0) GO TO 1
C
C        TOP (Y = YF)
C
      X1          = X1 + (YF - Y1) * (X2 - X1) / (Y2 - Y1)
      Y1          = YF
      GO TO 1
C
C        IF WE REACH HERE, THE LINE FROM (X1,Y1) TO (X2,
C             Y2)
C        IS VISIBLE.
C
    6 CALL PLOT (X1, Y1, 3)
      CALL PLOT (X2, Y2, 2)
      IF (S1) CALL SYMBOL (X1, Y1, 0.105, LS, 0.0, -1)
      IF (S2) CALL SYMBOL (X2, Y2, 0.105, LS, 0.0, -1)
      RETURN
      END
```

```
FUNCTION    KODE (X, Y)
COMMON  /FRAME/  XF, YF
KX          = 0
IF (X .LT. 0.0) KX = 1
IF (X .GT. XF) KX = 2
KY          = 0
IF (Y .LT. 0.0) KY = 4
IF (Y .GT. YF) KY = 8
KODE        = KX + KY
RETURN
END
```